

Smart Aid

Patient
Tracking
system

TABLE OF CONTENTS

- Introduction
- Components Used
- Architecture
- Code used in Project
- Screen shot of Smart Aid
- Future Development

PROJECT BY

Sam Thapa

Anton Andrei Bagasao

Introduction

Smart Aid is a Proof of concept design for an IOT wearable device that keeps track of user's vital signs designed to provide a faster and more accurate emergency response for users especially for time sensitive emergencies like cardiac arrests and/or seizures. Typically, emergency calls are made once an arrest has occurred and witnessed, but with Smart Aid, not only will it be able to detect abnormalities in vital signs, it can also automatically notify not only family members, but EMTs as well, of the patient's condition and location, even before signs of a cardiac arrest is prominently visible, saving valuable time usually taken from making an emergency call and in the process, increasing the chance of survival of the patient. Smart Aid can also share the vital signs of the Patient to the EMTs and Hospitals letting them be able to properly prepare for the patient's immediate care way before they arrive. According to one study, reducing the response time by 1 minute increases survival rate of an arrest by 24% (<http://emj.bmj.com/content/28/8/703>). Other emergencies such as Car Accidents and natural disasters like earthquakes can also benefit from this feature of Smart Aid. During the development of this project, we will be able to establish a more concrete scope for the features we would implement.

COMPONENTS USED

Smart Aid is built using various electric components. Some of them are

- **Arduino Uno**
- **Sparkfun PPG sensor**
- **Sparkfun ESP8266 thing board**
- **Adafruit GPS/GSM breakout**
- **Push Button**

Arduino Uno is the basic building component that connects all the components like sensors and actuators.

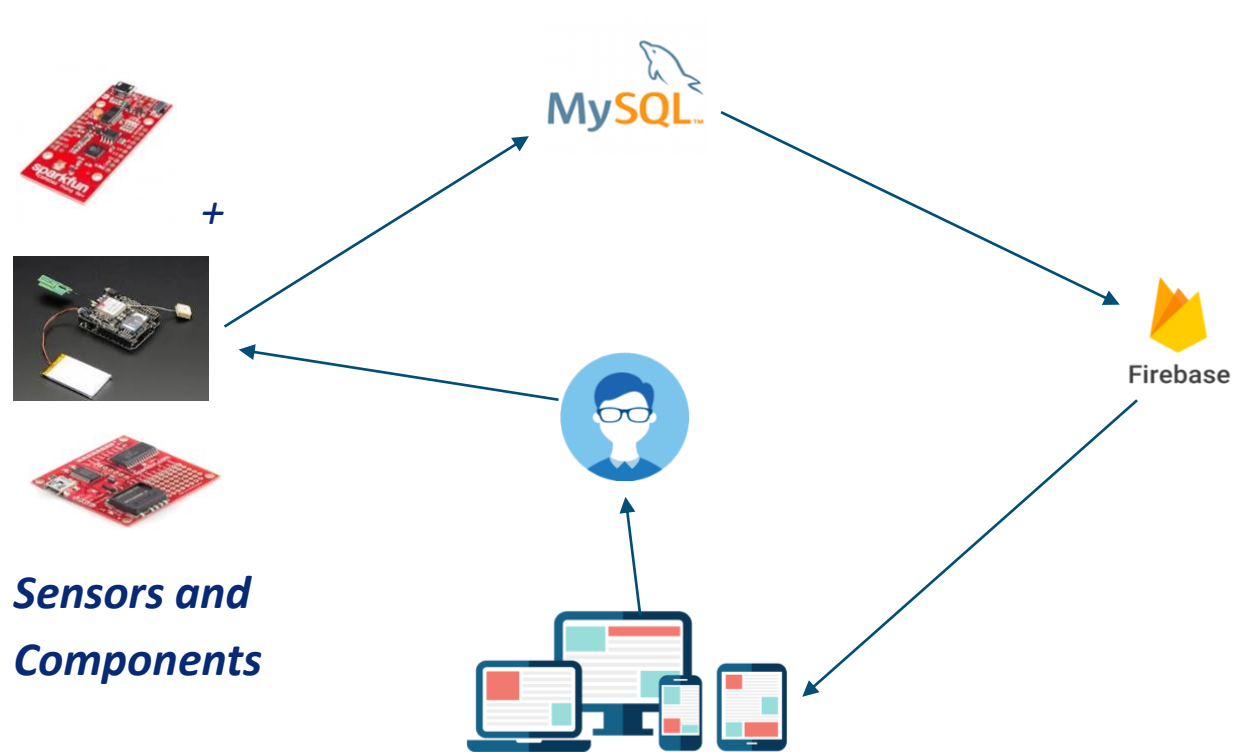
Sparkfun PPG sensor in our project, calculates the BPM of a person. We do some data analysis in our Arduino code and after the relevant value of BPM is found. We then send that data to our backend system specifically Google Firebase.

Sparkfun ESP8266 thing board is used to send the BPM of a person to the server periodically.

Adafruit GPS/GSM breakout is used to send the location of a person periodically to our web server.

Push Button is used in our project so that a person whenever feels uncomfortable and need help, he/she can push that panic button to send emergency text to their relatives. In that emergency text, "HELP I JUST PUSHED THE BUTTON, please track me at samthapa.me/tracker.php".

ARCHITECTURE OF SMART AID



WEB APPLICATION

All the data we get from our sensors; we send them to a web server using GET REQUEST in Arduino. Adafruit Fona has its own Library to do GET request. Adafruit FONA sends the location to web server, and Sparkfun Wi-Fi module sends the BPM of the patient to web server again using the GET request. All the data are send to our MYSQL database. Since MYSQL do not support real time data base,

We needed a real time database, Google Firebase which is very poplar these days. So to achieve the real time data streaming, in our web server, we have a MYSQL handler that acts as a middle man between Arduino and Google Firebase.

From the Google Firebase Database, we fetch the necessary data Using Front end technologies HTML/CSS/JAVASCRIPT. Every time a new data is pushed to Firebase database, users can see the change in real time, that means without even clicking or refreshing the browser.

Not only that, our system also uses Twilio API to send emergency test to the relatives of patient whenever there has been high reading of BPM. In our client side, we do AJAX request every 10 min and read the BPM and send emergency text if necessary

CODE USED

I) Adafruit Code

```
void loop() {
  char LAT1[10]; //string of lat and long unparsed & overflowing bound
  char LAT[10];
  char LONGI[10];

  float latitude, longitude; //variables to hold initial GPS readings
  boolean gps_success = fona.getGPS(&latitude, &longitude);
  if (gps_success) {
    Serial.print("GPS lat:");
    Serial.println(latitude, 6);
    Serial.print("GPS long:");
    Serial.println(longitude, 6);
    //send sms
    char message[141];
    char buffer1[10];
    char buffer2[10];
    String longitu = dtostrf(longitude, 10, 7, buffer1);
    String latitu = dtostrf(latitude, 10, 7, buffer2);
    for(int i = 0; i < 9; i++) {
      LAT[i] = LAT1[i];
    }
    LAT[9] = '\0'; //truncate array at last desired value

    if (buttonPressed == true)
    {
      char message[141];
      sprintf(message, " I JUST PUSHED MY PANIC BUTTON! I NEED HELP NOW! http://www.samthapa.me/tracker.php");
      char sendto[13] = "+15628815062";
      fona.sendSMS(sendto, message); //sends the message via SMS
      buttonPressed = false;
    }

    uint16_t statuscode;
    int16_t length;
    char urlString[70];
    String url = "http://www.samthapa.me/test1.php?";
    url += "lat=";
    url += latitu;
    url += "&longi=";
    url += longitu;
    url += "&date=2018-2-22";

    url.toCharArray(urlString, url.length());

    if (!fona.HTTP_GET_start(urlString, &statuscode, (uint16_t *)&length)) {
      Serial.println("Failed to send to MYSQL database");
    }
    while (length > 0) {
      while (fona.available()) {
        char c = fona.read();
        // Serialwrite is too slow, we'll write directly to Serial register!
      }
    }
  }
}

#ifdef __AVR_ATmega328P__ || defined(__AVR_ATmega168__)
```

II) WIFI module code

```
}
void sendtoMyServer(float bpm){
    // Use WiFiClient class to create TCP connections
    WiFiClient client;
    const int httpPort = 80;
    if (!client.connect(host, httpPort)) {
        String url = "http://samthapa.me/test1.php?lat=33.04933&longi=-134.08981&datee=2001-2-2&BPM="+ String(bpm);
        Serial.print("Requesting URL: ");
        Serial.println(url);
        client.print(String("GET ") + url + " HTTP/1.1\r\n" +
            "Host: " + host + "\r\n" +
            "Connection: close\r\n\r\n");
        unsigned long timeout = millis();
        while (client.available() == 0) {
            if (millis() - timeout > 5000) {
                Serial.println(">>> Client Timeout !");
                client.stop();
                return;
            }
        }
    }
}

void loop() {
    |
    long irValue = particleSensor.getIR();
    if (checkForBeat(irValue) == true)
    {
        long delta = millis() - lastBeat;
        lastBeat = millis();
        beatsPerMinute = 60 / (delta / 1000.0);
        if (beatsPerMinute < 255 && beatsPerMinute > 20)
        {
            rates[rateSpot++] = (byte)beatsPerMinute; //Store this reading in the array
            rateSpot %= RATE_SIZE; //Wrap variable

            //Take average of readings
            beatAvg = 0;
            for (byte x = 0 ; x < RATE_SIZE ; x++)
                beatAvg += rates[x];
            beatAvg /= RATE_SIZE;
        }
    }

    if (irValue < 50000)
        Serial.println(" No finger?");
    Serial.println(beatsPerMinute);
    sendtoMyServer(beatsPerMinute);
}
}
```

III) PHP code

```
11 // Create connection
12 $conn = mysqli_connect($servername, $username, $password, $dbname);
13 // Check connection
14 if (!$conn) {
15     die("Connection failed: " . mysqli_connect_error());
16 }
17
18 $lat = $_GET['lat'];
19 $longi = $_GET['longi'];
20 $datee = $_GET['datee'];
21 $bpm = $_GET['BPM'];
22
23 $DEFAULT_URL = 'https://tracker-1524239762376.firebaseio.com/';
24 $DEFAULT_TOKEN = 'q5z90k24rAOpkCjBmj8quDsrwxgfWHZ7e2klqbJZ';
25 $DEFAULT_PATH = '/Locations';
26
27 // reference for BPM root
28 $BPM_PATH = '/BPM';
29
30 // to store BPM
31 $bpmData = array(
32     'bpm' => $bpm
33 );
34
35 $fb = new firebase($DEFAULT_URL, $DEFAULT_TOKEN);
36 $new_response = $fb->push($BPM_PATH,$bpmData );
37
38 // only insert when everything in uri is set
39 if ( isset($_GET['lat']) && isset($_GET['longi']) && isset($_GET['datee']) )
40 {
41
42     //save to firebase
43
44     //to store lat long
45     $_devicestatus= array(
46         'latitude' => $lat + 0.0,
47         'longitude' => $longi + 0.0,
48     );
49
50     $response = $fb->update($DEFAULT_PATH, $_devicestatus);
51     sleep(2);
52     // save to mysql
53
54     $sql = "INSERT INTO Gps_Tracker (lat, longi, datee)
55     VALUES ($lat, $longi, $datee)";
56
57     if (mysqli_query($conn, $sql)) {
58         echo "Record Saved to database successfully";
59     } else {
60         echo "Error: " . $sql . "<br>" . mysqli_error($conn);
61     }
62 }
63 else{
64     echo "Please pass all parameters";
65 }
66
67 mysqli_close($conn);
```


IV) Front end code

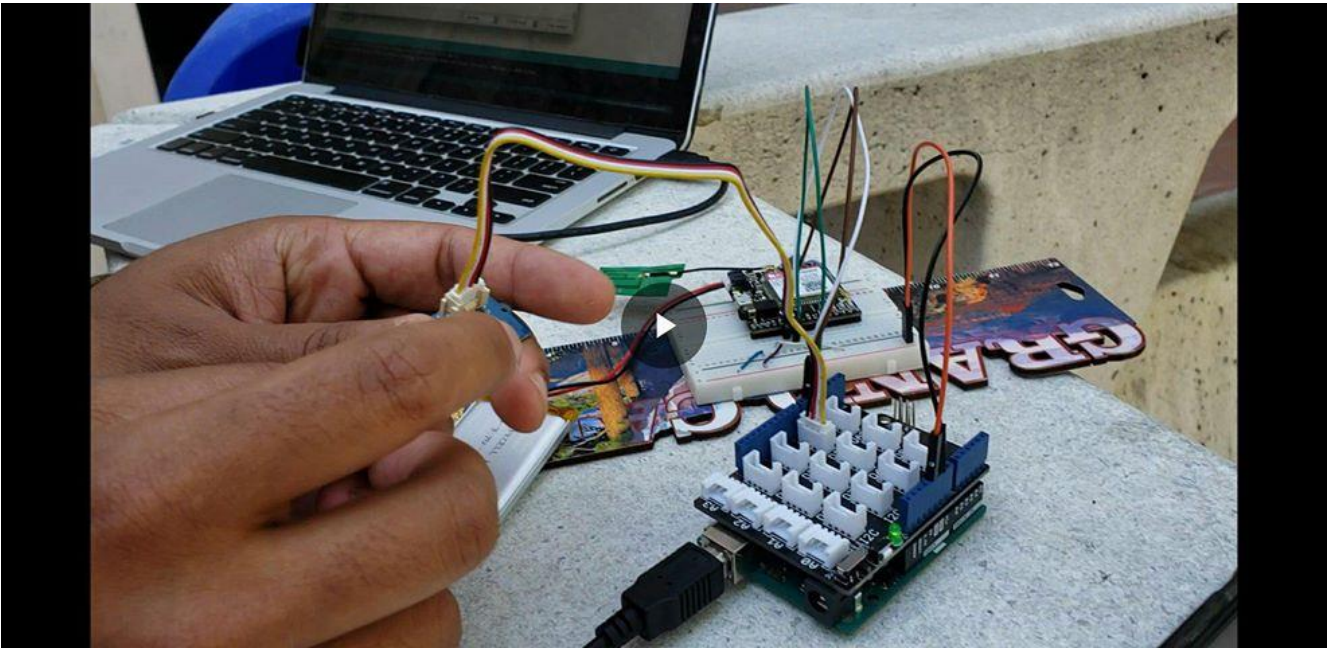
```
80 <script>
81
82     var lat;
83     var longi;
84     var Xdata = []; // to store values like [1,2,3,4, 6,7]
85     var Ydata = []; // to store values like [88, 89,73, 80]
86     var send = false;
87     var bpmRef = firebase.database().ref().child('BPM');
88     bpmRef.on("value", gotData, errData);
89     function gotData(data){
90         var bpms = data.val();
91         var keys = Object.keys(bpms);
92         for ( var i = 0; i < keys.length; i++)
93         {
94             var k = keys[i];
95             var bpm = bpms[k].bpm;
96             Xdata.push(i+1);
97             Ydata.push(bpm);
98         }
99         plotdata(Xdata, Ydata);
100     }
101
102     function plotdata(X, Y){
103
104         if (Y[Y.length-1] > 100 ){
105             twilioAjax();
106         }
107         document.getElementById("p1").innerHTML = Y[Y.length-1];
108         Plotly.plot('graph', [{
109             x: X,
110             y: Y
111         }], {
112             sliders: [{
113                 pad: {
114                     t: 30
115                 },
116                 currentvalue: {
117                     xanchor: 'right',
118                     prefix: 'color: ',
119                     font: {
120                         color: '#888',
121                         size: 10
122                     }
123                 },
124                 steps: [{
125                     label: 'red',
126                     method: 'restyle',
127                     args: ['line.color', 'red']
128                 }, {
129                     label: 'green',
130                     method: 'restyle',
131                     args: ['line.color', 'green']
132                 }
133             ]
134         }
```

```

1  function twilioAjax(){
2
3
4
5      $.ajax({
6          type: 'POST',
7
8          url: 'https://api.twilio.com/2010-04-01/Accounts/' + SID + '/Messages.json',
9          data: {
10              "To" : "+15628815062",
11              "From" : "+19093031617",
12              "Body" : "Alert !!! Patient has high chance of heart attack"
13          },
14          beforeSend: function (xhr) {
15              xhr.setRequestHeader ("Authorization", "Basic " + btoa(SID + ':' + Key));
16          },
17          success: function(data) {
18              console.log(data);
19          },
20          error: function(data) {
21              console.log(data);
22          }
23      });
24
25      setTimeout(twilioAjax, 100000);
26
27  }
28  function errData(err){
29      console.log("Error occured ");
30      console.log(err);
31  }
32  function initMap(){
33      var marker, i;
34      var rootRef = firebase.database().ref().child('Locations');
35
36      rootRef.on("value", function(snapshot) {
37          lat = snapshot.child('latitude').val();
38          longi = snapshot.child('longitude').val();
39          console.log(snapshot.child('latitude').val());
40          var map = new google.maps.Map(document.getElementById('map'), {
41              zoom: 14,
42              center: new google.maps.LatLng(lat, longi),
43              mapTypeId: google.maps.MapTypeId.ROADMAP
44          });
45          var infowindow = new google.maps.InfoWindow();
46
47
48          marker = new google.maps.Marker({
49              position: new google.maps.LatLng(lat, longi),
50
51              map: map
52          });
53
54          }, function (error) {
55              console.log("Error retriving data: " + error.code);
56          });
57      }
58  </script>
59  <!-- Replace the value of the key parameter with your own API key. -->

```

SCREEN SHOT OF SMART AID

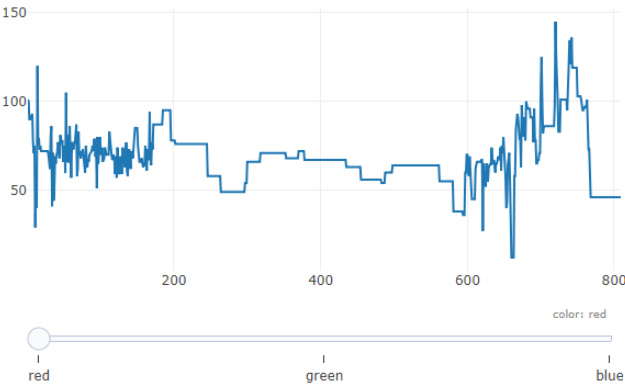


GPS TRACKER




BPM
46.00

DATA VISUALIZATION



Sent from your Twilio trial account -
Alert !!! Patient has high chance of
heart attack

Sent from your Twilio trial account -
Alert !!! Patient has high chance of
heart attack

Sent from your Twilio trial account -
Alert !!! Patient has high chance of
heart attack

Sent from your Twilio trial account -
Alert !!! Patient has high chance of
heart attack

Sent from your Twilio trial account -
Alert !!! Patient has high chance of
heart attack

Sent from your Twilio trial account -
Alert !!! Patient has high chance of
heart attack

Sent from your Twilio trial account -
Alert !!! Patient has high chance of
heart attack

samthapa.me >

I JUST PUSHED MY PANIC
BUTTON! I NEED HELP NOW!

[Tap to Load Preview](#)

samthapa.me >

I JUST PUSHED MY PANIC
BUTTON! I NEED HELP NOW!

[Tap to Load Preview](#)

samthapa.me >

I JUST PUSHED MY PANIC
BUTTON! I NEED HELP NOW!

[Tap to Load Preview](#)



Text Message



FUTURE DEVELOPMENT

Throughout the development of the Smart Aid project, we were able to pinpoint several key features we could add for future development that can be categorized between the SmartAid device itself and user interaction with the device.

First and foremost, we would like to improve on the data analysis to more accurately and effectively detect heart attacks and strokes. There are a couple of medical journals that details machine learning models that shows promising results hitting up to 97% success rate in detecting strokes.

Another area of improvement is the wearability of our device moving from a modular design of an Arduino-based device, having separate

modules and breadboards to implement sensors and features, to an integrated circuit design. One possible way we could implement this is with the use of FPGA programming.

Integrating more sensors like for body temperature and step counter to get a more detailed and well-rounded data is another area for future development that we have identified and could help in diagnosing a user's current health status.

For user experience, we could develop SmartAid to support multiple users, which means tracking of unique ID's per device, retooling our tracker page to support multiple users with unique logins and user data.